



**I. REAL PARTY IN INTEREST**

The subject application is owned by National Instruments Corporation, a corporation organized and existing under and by virtue of the laws of the State of Delaware, and having its principal place of business at 11500 N. MoPac Expressway, Bldg. B, Austin, Texas 78759-3504.

## **II. RELATED APPEALS AND INTERFERENCES**

No related appeals or interferences are known which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

### **III. STATUS OF CLAIMS**

Claims 1-88, 99, and 116 are canceled. Claims 89-98, 100-115, and 117-123 stand rejected and are the subject of this appeal. A copy of claims 89-98, 100-115, and 117-123 incorporating entered amendments and as on appeal, is included in the Claims Appendix hereto.

#### **IV. STATUS OF AMENDMENTS**

All amendments have been entered. No amendments to the claims have been filed subsequent to the rejection in the Office Action of August 5, 2005. The Claims Appendix hereto reflects the current state of the claims.

## V. SUMMARY OF THE CLAIMED SUBJECT MATTER

The subject matter recited in the present claims relates generally to the field of graphical programming. In traditional text-based programming, a program developer creates a program by writing textual program statements in a text-based programming language. The graphical programming paradigm differs from traditional text-based programming in that instead of writing text-based source code, the program developer creates a block diagram that visually indicates and defines the functionality of the program. More particularly, the program developer selects various function nodes or icons and interconnects the nodes, e.g., by drawing lines or wires between them such that the interconnected nodes visually indicate functionality of the program, e.g., visually indicate a function or process performed by the program. Thus, the term “graphical program” is used to refer to a program that comprises a plurality of interconnected nodes that visually indicate functionality of the program. (*See Summary of the Invention, p. 9; Figures 15-17 and accompanying text on pp. 40-43.*)

More particularly, the subject matter recited in the present claims relates to configuring a graphical user interface (GUI) element in a graphical program to receive and display data from a data source. According to the method of claim 89, a first GUI element for a graphical program is displayed on a display of a first computer system. Examples of GUI elements that may be used in a graphical program include graphs, text boxes, check boxes, knobs, etc. Other examples include GUI elements that are specialized for displaying data for instrumentation or measurement applications, such as thermometers, tanks, meters, waveform charts, etc. (*See Figures 1 and 2; block 200 of Figure 6; also Specification, p. 1, lines 8-21 and p. 27, line 21 – p. 28, line 24*)

The method further comprises receiving user input specifying a data source with which to associate the first GUI element. In response to receiving the user input specifying the data source, the first GUI element is automatically configured to receive and display data from the specified data source. In other words, the program developer is not required to implement the functionality of the first GUI element receiving and displaying the data from the specified data source, but rather the first GUI element is

automatically configured to perform this functionality. (*See blocks 202 and 204 of Figure 6; also Specification, p. 28, line 25 – p. 29, line 23*)

The method further comprises the first computer system receiving data from the specified data source, wherein the data includes information specifying a first data type of the data, and automatically determining that the first GUI element cannot display data of the first data type. For example, the program developer may have included an inappropriate first GUI element in the graphical program because the first GUI element is incompatible with data of the first data type, i.e., cannot display data of the first data type.

The method further comprises automatically substituting a second GUI element for the first GUI element, wherein the second GUI element can display data of the first data type, and displaying the received data from the specified data source on the second GUI element. (*See Summary of the Invention, p. 7, lines 1-7; p. 45, lines 13-17; and blocks 604 and 606 of Figure 29*)

Independent claim 105 is a memory medium claim analogous to the method claim 89 and recites similar limitations as claim 89.

Independent claim 106 recites a method similar to the method of claim 89. According to the method of claim 106, a first GUI element for a graphical program is displayed on a display of a first computer system, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program. User input specifying a data source with which to associate the first GUI element is received. In response to receiving the user input specifying the data source, the first GUI element is automatically configured to receive and display data from the specified data source. The first computer system receives data from the specified data source, wherein the data includes information specifying a first data type of the data, similarly as described above with respect to claim 89.

The method further comprises automatically determining if the first GUI element can display data of the first data type. If the first GUI element cannot display data of the first data type then an invalid condition is indicated. For example, the invalid condition may be indicated by displaying text and/or graphical information in the GUI of the

graphical program to indicate that the first GUI element cannot display the data from the specified data source. (*See p. 6, lines 25-29; p. 44, line 22 – p. 45, line 12*)

Independent claim 122 is a memory medium claim analogous to the method claim 106 and recites similar limitations as claim 106.



## **VI. GROUND OF REJECTION TO BE REVIEWED ON APPEAL**

Claims 89-98, 100-101, 103-115, 117-118, and 120-123 stand rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,903,728 to Semenzato (hereinafter “Semenzato”).

Claims 102-104 and 119-121 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Semenzato in view of U.S. Patent No. 5,692,213 to Goldberg et al. (hereinafter “Goldberg”).

## VII. ARGUMENT

### Section 102(b) Rejection

Claims 89-98, 100-101, 103-115, 117-118, and 120-123 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Semenzato. Appellant respectfully traverses this rejection for the following reasons.

### Independent Claims 89, 105, 106, and 122

Independent claim 89 recites as follows:

89. (Previously Presented) A method for configuring a graphical user interface (GUI) element to subscribe to a data source, the method comprising:

displaying a first GUI element for a graphical program on a display of a first computer system, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program;

receiving user input specifying a data source with which to associate the first GUI element;

in response to said receiving user input, automatically configuring the first GUI element to receive and display data from the specified data source;

the first computer system receiving data from the specified data source, wherein the data includes information specifying a first data type of the data;

automatically determining that the first GUI element cannot display data of the first data type;

automatically substituting a second GUI element for the first GUI element, wherein the second GUI element can display data of the first data type; and

displaying the received data from the specified data source on the second GUI element. (*Emphasis added*)

Thus, the method comprises in pertinent part, “displaying a first GUI element for a graphical program on a display of a first computer system, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program”. Regarding the interpretation of this portion of the claim, the Examiner states on p. 2 of the Office Action of August 5, 2005 that,

“Semenzato presents a GUI element, specifically that of a plug-in associated with a browser application. Such a browser application is considered a graphical program, since as known in the art, it comprises graphical elements to visually present information to the user. Additionally, such a browser is considered to comprise a plurality of interconnected nodes which visually

indicate functionality of the graphical program, since as known in the art, a graphical display is generally comprised of interconnected nodes. When specifying the rudimentary visual characteristics of a graphical display instance, graphics primitives are applied to specify points, i.e., nodes, which are interconnected to form the graphical objects, such as lines and polygons, composing the display (for example, see the included Chapters 1 and 2 of “Computer Graphics: Principles and Practice,” with particular regard to pages 26-31). The window of the browser of Semenzato, for example, is thus understood to be comprised of four nodes, one at each corner, which are interconnected to form the rectangle delineating the window. Accordingly, given the broadest, most reasonable interpretation of a graphical program comprising a plurality of interconnected nodes which visually indicate functionality of the graphical program, the browser of Semenzato is considered such a graphical program.”

However, as described in Section 2111 of the MPEP, pending claims must be given the broadest reasonable interpretation consistent with the specification. The broadest reasonable interpretation of the claims must also be consistent with the interpretation that those skilled in the art would reach. Appellant submits that the Examiner’s interpretation of what is meant by a “graphical program” and “a plurality of interconnected nodes which visually indicate functionality of the graphical program” is not consistent with the specification. Appellant also submits that in light of the specification, those skilled in the art would not interpret the claim terminology as the Examiner did above.

As described in the Summary of the Invention on p. 9,

“In one embodiment, the programs which include the GUI elements may be graphical programs. Graphical programming has become a popular programming paradigm, especially among scientists and engineers. In creating a graphical program, a developer may include a plurality of graphical nodes and other graphical elements on block diagram and connect the nodes and elements such that the resulting diagram graphically represents the functionality of the graphical program. The graphical nodes or icons may be connected using one or more of data flow, control flow, and/or execution flow techniques. Many developers find graphical programming to be a more intuitive and user friendly model for developing a program, e.g., as opposed to using a traditional text-based programming language, such as C, C++, Basic, etc. Thus, one embodiment of the present invention further enhances the user-friendliness of graphical programming by enabling a developer to easily configure a data connection for a GUI element on a graphical program’s graphical user interface panel.”

Thus, graphical programming differs from traditional text-based programming in that instead of creating textual program statements in a text-based programming language, the program developer creates a graphical block diagram that defines the functionality of the graphical program. In creating the block diagram, the developer includes a plurality of nodes on the block diagram and interconnects the nodes so that the interconnected nodes visually indicate functionality of the graphical program, e.g., visually indicate a function or process performed by the graphical program.

Appellant also notes that U.S. Patent Application No. 09/518,492, titled, “System and Method for Programmatically Creating a Graphical Program,” which is incorporated by reference in the present application, provides further information regarding what is meant by a graphical program throughout the disclosure, and in particular, in the Description of the Related Art section on pp. 1-5. For example, on p. 4, this patent application states that, “During creation of the graphical program, the user selects various functions that accomplish his desired result and connects the function icons together... The assembled graphical program may then be compiled or interpreted to produce machine language that accomplishes the desired method or process as shown in the block diagram.” (Note that the term “icon” is used interchangeably with the term “node” in this context.)

The present application also provides several drawings (Figures 15-17) illustrating exemplary graphical programs. Each of these drawings illustrates a plurality of interconnected nodes which visually indicate functionality of the respective graphical program. The accompanying text on pp. 40-43 of the specification describes the operation of these graphical programs and the functions performed by various nodes included in the graphical programs.

According to the Examiner’s interpretation of a graphical program as a program that “comprises graphical elements to visually present information to the user,” any program with a graphical user interface would be considered a graphical program. However, as described above, this is not at all what is meant by the term “graphical program” in the present application, and is not consistent with the specification, and is not how the term would be interpreted by those skilled in the art. Furthermore, the Examiner’s interpretation of a plurality of interconnected nodes in a graphical program as

the four corners of a browser window is not at all consistent with the specification and would not be interpreted as such by those skilled in the art.

Appellant respectfully submits that if the claims are interpreted reasonably and consistently with the specification then it is clear that Semenzato does not teach the concept of a graphical program that comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program, as recited in claim 89.

Appellant also submits that Semenzato does not teach several other aspects of the subject matter recited in claim 89. For example, the claim recites, “displaying a first GUI element for a graphical program on a display of a first computer system” and “automatically substituting a second GUI element for the first GUI element”. The Examiner has attempted to equate the method of claim 89 with a method of using plug-ins to display data in a web browser, such as taught by Semenzato. However, Semenzato does not teach displaying a first GUI element and then substituting a second GUI element for the first GUI element.

As is known to those generally familiar with web browser technology, when a browser starts, it checks the available plug-ins, reads resources from each plug-in file to determine the MIME types for that plug-in, and registers each plug-in. When the browser encounters data of a MIME type registered for a particular plug-in, it invokes the plug-in so that the plug-in can process the data. As part of its operation, the plug-in may display a GUI element within the browser window. However, the GUI element associated with the plug-in does not substitute for another GUI element, but is simply displayed within the browser.

The Examiner has equated the “second GUI element” recited in claim 89 with the GUI element associated with the plug-in. It is not clear from the Examiner’s explanation for the rejection of claim 89 what exactly is considered to be the “first GUI element”. However, claim 91 recites, “The method of claim 89, wherein the first GUI element is automatically configured without user programming.” In the rejection of claim 91, it is clear that the Examiner has equated the first GUI element with the browser window. The Examiner writes that, “it is understood that the only user input involved in configuring a window of a browser to receive and display data from a data source may be that of selecting a hyperlink specifying the data source, as is known in the art. The window is

thus considered to be automatically configured without user programming and without user input specifying program code.”

Thus, since the Examiner has interpreted the first GUI element to be the browser window and the second GUI element to be the GUI element associated with the plug-in, in order to teach the subject matter recited in claim 89 it would be necessary for Semenzato to teach that the GUI element associated with the plug-in substitutes for the browser window. However, Semenzato does not teach this, and this is not how plug-ins generally operate. Instead, the GUI element associated with the plug-in is simply displayed within the browser window. The browser window is still displayed and remains operational.

Furthermore, Appellant submits that the Examiner’s interpretation of the first GUI element as a browser window is erroneous. It is clear from the specification that the first GUI element is not itself a window, but is an element displayed on or within a window or panel. For example, p. 1 of the specification states that, “Figure 2 (prior art) illustrates additional examples of GUI elements useful for instrumentation or measurement applications, e.g., a thermometer, an LED, a meter, a waveform chart, a tank, etc. Other types of GUI elements that may be included on a graphical user interface panel include text boxes, check boxes, etc.” Also, p. 5 states that, “Typically, the developer creates a graphical user interface panel and places various GUI elements on the panel, e.g., by choosing the GUI elements from a palette, from menu options, from a dialog box, etc.” Throughout the specification, it is made clear that a GUI element refers to an element on or within a panel or window, not to a window itself.

Thus, for at least the reasons set forth above, Appellant respectfully submits that the subject matter of claim 89 is not anticipated by Semenzato, and thus, claim 89 is patentable over Semenzato. Since claims 90-98 and 100-101 are dependent upon claim 89, Appellant submits that these claims are also patentable over Semenzato, for at least this reason.

Independent claims 105, 106, and 122 recite similar elements of, “displaying a first GUI element for a graphical program on a display of a first computer system, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program”. Thus, Appellant respectfully

submits that these claims, and the claims respectively dependent thereon, are also patentable over Semenzato, for reasons similar to those set forth above.

### **Claims 93 and 110**

Claims 93 and 110 are separately patentable because Semenzato does not teach the additional limitation recited in these claims of, “wherein said receiving user input specifying the data source comprises receiving user input via a user interface dialog box.” As is known to those skilled in the art of programming, a dialog box refers to a modal window. A dialog box or other modal window must be closed (e.g., by clicking on an “Ok” or “Cancel” button) before the user can interact with other windows of the application. A dialog box is typically a pop-up window which the application temporarily displays to solicit a response from the user. Appellant respectfully submits that the Examiner is incorrect in the assertion that, “the browser window displaying a hyperlink is considered a dialog box, like recited in claim 93, as it is a window displayed by the browser to solicit a response from the user.” A browser window is not a modal dialog box window and is not displayed specifically to solicit a response from the user. Appellant thus submits that Semenzato does not teach the limitation recited in claims 93 and 110.

### **Claim 96 and 113**

Claims 96 and 113 are separately patentable because Semenzato does not teach the additional limitation recited in these claims of, “wherein said user input specifying the data source is received during development of the graphical program.” As discussed above, the Examiner has equated the graphical program with the browser program. Thus, in order to teach this limitation, Semenzato would have to teach receiving the user input specifying the data source during development of the browser program.

Semenzato contains teaching regarding the development of a plug-in program, and specifically, regarding the debugging of a plug-in program. However, Semenzato contains no teaching regarding the development of the browser program, which is developed separately from the plug-in program. More specifically, Semenzato does not teach performing the combination of elements recited in claims 89 and 106 during

development of a browser program. Appellant thus submits that Semenzato does not teach the limitations recited in claims 96 and 113.

### **Claims 98 and 115**

Claims 98 and 115 are separately patentable because Semenzato does not teach the additional limitation recited in these claims of, “executing the graphical program after said automatically configuring”. Appellant notes that claims 89 and 106 recite, “in response to said receiving user input, automatically configuring the first GUI element to receive and display data from the specified data source”. Thus, the automatic configuration of the first GUI element is performed after receiving the user input specifying the data source (since the automatic configuration is performed in response to the user input). Thus, in claims 98 and 115, the graphical program is executed after receiving the user input specifying the data source, since the graphical program is executed after the automatic configuration of the first GUI element, and the automatic configuration of the first GUI element is performed after (in response to) receiving the user input specifying the data source.

The Examiner has interpreted the graphical program as a browser and has interpreted receiving the user input specifying the data source as the browser receiving user input specifying a URL. In order to teach the limitation recited in claims 98 and 115, the browser would have to be executed after receiving the user input specifying the URL. However, this is not possible because in order for the browser to receive the user input specifying the URL, the browser must already be executing. Thus, Semenzato does not teach the limitations recited in claims 98 and 115.

### **Claim 123**

Claim 123 is separately patentable because Semenzato does not teach the additional limitations recited in this claim. Claim 123 recites the limitations of, “wherein said displaying the first GUI element, said receiving the user input specifying the data source, said automatically configuring the first GUI element, said first computer system receiving data, said automatically determining, said automatically substituting the second



GUI element, and said displaying the received data are performed during creation of the graphical program.”

The Examiner has interpreted the graphical program as a browser program. Thus, in order to teach the limitations of claim 123, Semenzato would have to teach that the above-recited steps are performed during creation of the browser program. As noted above with reference to claims 96 and 113, Semenzato contains teaching regarding the debugging of a plug-in program, which is a separate program from the browser program. However, Semenzato contains no teaching regarding the debugging of a browser program.

Moreover, Appellant notes that in claim 123, the phrase “creation of the graphical program” refers to the edit-time act of creating the graphical source code for the graphical program, e.g., by assembling the plurality of interconnected nodes in the block diagram of the graphical program. In contrast, the act of debugging a plug-in program as described in Semenzato involves executing the plug-in program in the context of a debugger after the plug-in program has been created (See Col. 2). Thus, Semenzato does not teach performing the above-recited steps during creation of a plug-in program because the plug-in program must be created before it can be executed in the context of a debugger.

### **Section 103(a) Rejection**

Claims 102-104 and 119-121 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Semenzato in view of Goldberg. Appellant notes that if an independent claim is non-obvious under 35 U.S.C. 103, then any claim depending therefrom is non-obvious. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988). Thus, since the independent claims have been shown to be patentably distinct from and non-obvious in view of the cited references, Appellant respectfully submits that dependent claims 102-104 and 119-121 are also patentably distinct and non-obvious.

Claim 103 recites the limitation of, “wherein the data is live data.” Claim 104 depends on claim 103 and recites the limitation of, “wherein the data comprises measurement data.” Measurement data refers to scientific or technical data acquired by a measurement device, e.g., where the measurement device is coupled to a sensor that

measures a physical property of a device or system under test. For example, p. 41 of the specification states that, “For example, a user may easily monitor real-time measurement data acquired by an instrument located in a remote laboratory.”

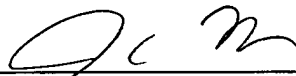
Goldberg does not teach the concept of live measurement data and thus does not teach the limitations recited in claim 104, and thus, claim 104 is non-obvious in view of the cited references. Inasmuch as claim 121 recites a similar limitation as claim 104, Appellant respectfully submits that claim 121 is also non-obvious in view of the cited references.

### VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 89-98, 100-115, and 117-123 was erroneous, and reversal of his decision is respectfully requested.

The Commissioner is authorized to charge the appeal brief fee of \$500.00 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5150-47600/JCH. This Appeal Brief is submitted with a return receipt postcard.

Respectfully submitted,



---

Jeffrey C. Hood  
Reg. No. 35,198  
ATTORNEY FOR APPLICANT(S)

Meyertons Hood Kivlin Kowert & Goetzel, P.C.  
P.O. Box 398  
Austin, TX 78767-0398  
Phone: (512) 853-8800

Date: January 9, 2006 JCH/JLB

## **IX. CLAIMS APPENDIX**

The following lists the claims as incorporating entered amendments, including claims 89-98, 100-115, and 117-123 as on appeal.

Claims 1-88. (Canceled)

89. (Previously Presented) A method for configuring a graphical user interface (GUI) element to subscribe to a data source, the method comprising:

displaying a first GUI element for a graphical program on a display of a first computer system, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program;

receiving user input specifying a data source with which to associate the first GUI element;

in response to said receiving user input, automatically configuring the first GUI element to receive and display data from the specified data source;

the first computer system receiving data from the specified data source, wherein the data includes information specifying a first data type of the data;

automatically determining that the first GUI element cannot display data of the first data type;

automatically substituting a second GUI element for the first GUI element, wherein the second GUI element can display data of the first data type; and

displaying the received data from the specified data source on the second GUI element.

90. (Previously Presented) The method of claim 89,

wherein the data source is located remotely from the first computer system and is coupled to the first computer system over a network, wherein the data source is specified using a URL; and

wherein said automatically configuring the first GUI element to receive and display data from the specified data source comprises:

automatically configuring the first GUI element to connect to the data source.

91. (Previously Presented) The method of claim 89, wherein the first GUI element is automatically configured without user programming.

92. (Previously Presented) The method of claim 89, wherein the first GUI element is automatically configured without user input specifying source code.

93. (Previously Presented) The method of claim 89,  
wherein said receiving user input specifying the data source comprises receiving user input via a user interface dialog box.

94. (Previously Presented) The method of claim 89, further comprising:  
the first GUI element performing said receiving and displaying data from the specified data source, wherein said performing is performed after said automatically configuring.

95. (Previously Presented) The method of claim 89,  
wherein the data source is comprised in a second computer system remotely located from the first computer system, wherein the first computer system is operable to connect to the second computer system over a network; and  
wherein said automatically configuring the first GUI element comprises automatically configuring the first GUI element to connect to the second computer and perform said receiving and displaying data from the specified data source.

96. (Previously Presented) The method of claim 89, wherein the first GUI element is associated with the graphical program;  
wherein said displaying the first GUI element comprises including the first GUI element in a user interface associated with the graphical program; and

wherein said user input specifying the data source is received during development of the graphical program.

97. (Previously Presented) The method of claim 96, wherein, during execution of the graphical program, the first GUI element is operable to perform said receiving and displaying data from the specified data source.

98. (Previously Presented) The method of claim 96, further comprising:  
executing the graphical program after said automatically configuring, wherein said executing the graphical program includes the first GUI element performing said receiving and displaying data from the specified data source.

99. (Canceled)

100. (Previously Presented) The method of claim 89, wherein the data source is one from the group consisting of:

an HTTP server;

an FTP server;

an OPC server;

an SNMP server;

a DataSocket server; and

a file.

101. (Previously Presented) The method of claim 89,  
wherein the data source is a remote data source associated with a remote computer; and

wherein said automatically configuring the first GUI element comprises automatically configuring the first GUI element to connect to the remote data source and receive and display data from the remote data source during program execution.

102. (Previously Presented) The method of claim 101, further comprising:

executing a computer program operable to publish live data to the remote data source;

wherein the first GUI element is operable to display the live data.

103. (Previously Presented) The method of claim 89,  
wherein the data is live data.

104. (Previously Presented) The method of claim 103,  
wherein the data comprises measurement data.

105. (Previously Presented) A computer-readable memory medium that stores program instructions for configuring a graphical user interface (GUI) element to subscribe to a data source, wherein the program instructions are computer-executable to implement:

displaying a first GUI element for a graphical program on a display of a first computer system, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program;

receiving user input specifying a data source with which to associate the first GUI element;

in response to said receiving user input, automatically configuring the first GUI element to receive and display data from the specified data source;

the first computer system receiving data from the specified data source, wherein the data includes information specifying a first data type of the data;

automatically determining that the first GUI element cannot display data of the first data type;

automatically substituting a second GUI element for the first GUI element, wherein the second GUI element can display data of the first data type; and

displaying the received data from the specified data source on the second GUI element.

106. (Previously Presented) A method for configuring a graphical user interface (GUI) element to subscribe to a data source, the method comprising:

displaying a first GUI element for a graphical program on a display of a first computer system, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program;

receiving user input specifying a data source with which to associate the first GUI element;

in response to said receiving user input, automatically configuring the first GUI element to receive and display data from the specified data source;

the first computer system receiving data from the specified data source, wherein the data includes information specifying a first data type of the data;

automatically determining if the first GUI element can display data of the first data type; and

indicating an invalid condition if the first GUI element cannot display data of the first data type.

107. (Previously Presented) The method of claim 106,

wherein the data source is located remotely from the first computer system and is coupled to the first computer system over a network, wherein the data source is specified using a URL; and

wherein said automatically configuring the first GUI element to receive and display data from the specified data source comprises:

automatically configuring the first GUI element to connect to the data source.

108. (Previously Presented) The method of claim 106, wherein the first GUI element is automatically configured without user programming.

109. (Previously Presented) The method of claim 106, wherein the first GUI element is automatically configured without user input specifying source code.



110. (Previously Presented) The method of claim 106,  
wherein said receiving user input specifying the data source comprises receiving  
user input via a user interface dialog box.

111. (Previously Presented) The method of claim 106, further comprising:  
the first GUI element performing said receiving and displaying data from the  
specified data source, wherein said performing is performed after said automatically  
configuring.

112. (Previously Presented) The method of claim 106,  
wherein the data source is comprised in a second computer system remotely  
located from the first computer system, wherein the first computer system is operable to  
connect to the second computer system over a network; and  
wherein said automatically configuring the first GUI element comprises  
automatically configuring the first GUI element to connect to the second computer and  
perform said receiving and displaying data from the specified data source.

113. (Previously Presented) The method of claim 106, wherein the first GUI  
element is associated with the graphical program;  
wherein said displaying the first GUI element comprises including the first GUI  
element in a user interface associated with the graphical program; and  
wherein said user input specifying the data source is received during development  
of the graphical program.

114. (Previously Presented) The method of claim 113, wherein, during execution  
of the graphical program, the first GUI element is operable to perform said receiving and  
displaying data from the specified data source.

115. (Previously Presented) The method of claim 113, further comprising:

executing the graphical program after said automatically configuring, wherein said executing the graphical program includes the first GUI element performing said receiving and displaying data from the specified data source.

116. (Canceled)

117. (Previously Presented) The method of claim 106, wherein the data source is one from the group consisting of:

- an HTTP server;
- an FTP server;
- an OPC server;
- an SNMP server;
- a DataSocket server; and
- a file.

118. (Previously Presented) The method of claim 106, wherein the first data source is a remote data source associated with a remote computer; and

wherein said automatically configuring the first GUI element comprises automatically configuring the first GUI element to connect to the remote data source and receive and display data from the remote data source during program execution.

119. (Previously Presented) The method of claim 118, further comprising: executing a computer program operable to publish live data to the remote data source;

wherein the first GUI element is operable to display the live data.

120. (Previously Presented) The method of claim 106, wherein the data is live data.

121. (Previously Presented) The method of claim 103,

wherein the data comprises measurement data.

122. (Previously Presented) A computer-readable memory medium that stores program instructions for configuring a graphical user interface (GUI) element to subscribe to a data source, wherein the program instruction are computer-executable to implement:

- displaying a first GUI element for a graphical program on a display of a first computer system, wherein the graphical program comprises a plurality of interconnected nodes which visually indicate functionality of the graphical program;

- receiving user input specifying a data source with which to associate the first GUI element;

- in response to said receiving user input, automatically configuring the first GUI element to receive and display data from the specified data source;

- the first computer system receiving data from the specified data source, wherein the data includes information specifying a first data type of the data;

- automatically determining if the first GUI element can display data of the first data type; and

- indicating an invalid condition if the first GUI element cannot display data of the first data type.

123. (Previously Presented) The method of claim 89, wherein said displaying the first GUI element, said receiving the user input specifying the data source, said automatically configuring the first GUI element, said first computer system receiving data, said automatically determining, said automatically substituting the second GUI element, and said displaying the received data are performed during creation of the graphical program.

**X. EVIDENCE APPENDIX**

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

**XI. RELATED PROCEEDINGS APPENDIX**

There are no related proceedings.